Implementation of Authenticated Encryption Algorithm Offset Code Book (OCB)

Levent Ertaul, Sravya K L, Nagaraju Sanka
CSU East Bay, Hayward, CA, USA.
levent.ertaul@csueastbay.edu, SL@horizon.csueastbay.edu, nsanka@horizon.csueastbay.edu

***Abstract*— Data protection is one of the most important aspects for all users personal and other details. With the Increasing use of network based mobile phones, the concern for data protection for mobile users are increased rapidly. On a daily basis data theft occurs in all system and applications either by accident or on purpose. This kind of data breaches puts the data of numerous users at risk, at the same time it provides more advantage to the competitors to access confidential data of others in the network. Nowadays, many of the users store their personal and confidential data in internet without any security. To avoid these kind of problems we developed an android application called data vault to store the user personal details and confidential details in a secured manner by using Authenticated encryption. Authenticated Encryption is a technique used to convert the normal data into encrypted data and this data can be accessed after the authentication process. Where OCB (Offset Codebook Mode) is a secured data authenticated encryption mode of operation which is used in the cryptographic block ciphers. In the developed android application, we are using OCB mode for authenticated encryption which is an efficient technique for storing the data securely in the cloud data storage. In the android application user can store their details in the cloud and those details are stored in encrypted by using the OCB and to retrieve their stored details user need to go through OTP authentication process. This technique helps us to avoid detrimental loss in the data. At the same time using OCB in the system provides more advantages to secure data.**

I. INTRODUCTION

OCB mode is mainly designed for the message authentication and providing privacy for data. Data security is one of the most important aspects to avoid unwanted issues in the internet. As well many of the tools and software's consider data backup is one of the most important thing in the potential for a computer's hardware crash or damage, power surge and any other disastrous. Anyhow, data backup provides more advantage to users to recover lost data immediately. So, users can use both data encryption techniques to avoid unauthorized users access and data backup techniques to avoid data lost in the application or system [1].

Authenticated-encryption techniques allow to share the key with other users which is used encrypt plain text into human unreadable format, and it aims to provide privacy and authenticity for the data.

The encryption algorithms tools plaintext, key and nonce to created encrypted data. At the same time the decryption process take key, encrypted data and nonce to generate original data from the encrypted data. If the key is not match with the encrypted data, then it produces invalid symbols in the system. OCM describes the Jutla's schemes. It also deals with the principal characteristics of IAPM, and additionally it provides a privacy-only mode. Anyhow it integrates the following features in the system:

- Arbitrary-length messages + minimal-length cipher texts.
- Nearly optimal number of block-cipher calls.
- Minimal requirements on nonce.
- Improved offset calculations.
- Single underlying key.

Combining various techniques and methods we can easily active the above properties in the OCB. Many earlier versions of this algorithm rejected due to the maximum possibilities of attacks and no security for the data from the theft. "Fragile" techniques helped to identify the possible attacks in the algorithm and advanced techniques to avoid those issues in the present algorithm [2] [3].

Materialization of the AE modes serves as two different systems to increase the confidentiality and authenticity of the system and in applications. Block cipher is used to enable confidentiality of the users in the system. At the same time, it uses MAC for the authenticity process in the system [4]. Anyhow, OCB mode integrates both block cipher and MAC appropriately as well cost computation is very low as the two individual systems.

OCB mode accepts various block cipher with different sizes such as 128, 192 and 256. Length t and authentication tag T both is an integer between 0 and n. with the default suggestion of t value as 64bits, which provides a strength to forge a reasonable cipher text with probability 2^-t. this process mode will takes the key, plaintext and nonce as an input for the encryption algorithm and generate output as nonce, cipher text and authentication tags to decrypt the original data [5].

Decryption algorithm also works as an encryption algorithm but it takes encrypted data to analyze the tag and plaintext, once it identified both tags are matching then the

decryption algorithm retrieve the plaintext from the encrypted data [5] [6].

OCB mode is a plaintext aware due to the use of message authentication code (MAC) in the authenticity process. This mode enables IND-CPA secure which provides additional futures to the system. IND-CPA provides sematic security against user entered plaintext attack, at the same time cipher text doesn't leak the information about plaintext. This mode helps to the system to secure data against stronger attack particularly semantic security with the combination of selected cipher text attack. Based on this process it provides satisfactory security guarantees in the system [7].

Figure 1 shows, detailed information about the authenticated encryption algorithm of OCB mode in the various system.

- nonce N.
- plaintext M,
- output of the system tag C// T.

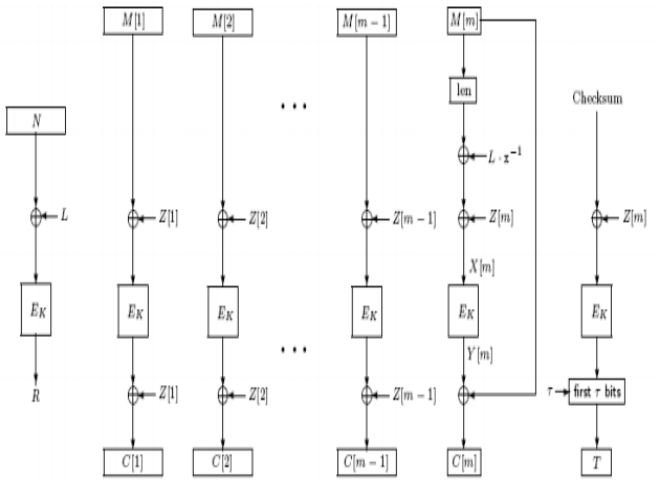

Figure 1: The authenticated encryption algorithm of OCB mode.

Offset Code Book (OCB) is a single algorithm; it provides various techniques with higher security for the data which is need by the most of application in the world. Comparing with other encryption algorithm OCB is twice as fast, and it use minimum key materials. The main objectives of OCB is creating minimal-length of ciphertext, and make all the process correctly in messages of arbitrary length. This algorithm is compiled very faster in the low resource components used system [7] [8].

This Section I describes about the concept of the authenticated encryption technique and OCB mode of operation and about the android application with this features. Section II tells us the requirement of the application developed for this algorithm. Section III, explains briefly about the proposed andriod application architecture, Section IV describes the encryption and decryption method followed by the OCB algorithm and features of OCB alogirthm is explained in Section V. Section VI consists of screen shots for each functionality in the android application for better understanding. Performance of each operation that processed in the data vault application are measured and explained in Section VII. Section VIII concluded bt stating benefits of this android application.

II. OFFSET CODE BOOK MODE ALGORITHM

A. *Algorithm for OCB Encryption*

This algorithm is used to encrypt user plaintext into human unreadable format [11] [12]. It takes minimum time to convert the plaintext into an encrypted data.

Algorithm OCB.EncK (N, M)

Partition M into M[1] ••• M[m]

L ← EK(0n)

R ← EK(N ⊕ L)

for i ← 1 to m do Z[i] = γi • L ⊕ R

for i ← 1 to m − 1 do C[i] ← EK(M[i] ⊕ Z[i]) ⊕ Z[i]

X[m] ← len(M[m]) ⊕ L • x−1 ⊕ Z[m]

Y [m] ← EK(X[m])

C[m] ← Y [m] ⊕ M[m]

C ← C[1] ••• C[m]

Checksum ← M[1] ⊕•••⊕ M[m − 1] ⊕ C[m] 0∗ ⊕ Y [m]

T ← EK(Checksum ⊕ Z[m]) [first τ bits]

return C ← C T

B. *Algorithm for OCB Decryption*

Algorithm OCB.DecK (N, C)

Partition C into C[1] ••• C[m] T

L ← EK(0n)

R ← EK(N ⊕ L)

for i ← 1 to m do Z[i] = γi • L ⊕ R

for i ← 1 to m − 1 do M[i] ← E−1

K (C[i] ⊕ Z[i]) ⊕ Z[i]

X[m] ← len(C[m]) ⊕ L • x−1 ⊕ Z[m]

Y [m] ← EK(X[m])

M[m] ← Y [m] ⊕ C[m]

M ← M[1] ••• M[m]

Checksum ← M[1] ⊕•••⊕ M[m − 1] ⊕ C[m] 0∗ ⊕ Y [m]

T ← EK(Checksum ⊕ Z[m]) [first τ bits]

if T = T then return M

else return Invalid

The above algorithms are written for OCB data encryption and decryption on the key space k. Anyhow, k is used as key space for the block cipher E.

C. *Key generation*

This algorithm selects random key to process it functionalities in the system. K □ (←┴R) K functionality is used in the block cipher. The "key" is K is provided for the operation of both data encryption and decryption [13].

D. *Key setup*

Mostly Key setup is associated with the process of block cipher enciphering and deciphering. L ← EK (0^n). m integrates the maximum number of n-bit blocks of message that are encrypted or decrypted. Let, μ←[⟦log⟧ _2 m]. L(0) ← L. i ∈ [1..μ]. Evaluate L(i) ← L(i−1)•x here the functionality use shift and a conditional XOR. Once the results are computed using the above conditions then the values are stored in the L(−1), L(0), L(1),..., L(μ) [14].

E. *Encryption*

To encrypt plaintext M ∈ {0, 1} ^∗ this process use K as a key and nonce N ∈ {0, 1} ^n, this functionality will generate cipher text c. Firstly it collects plaintext form the users as input and process that plaintext using the OCB encryption algorithm to generate encrypted data [20]. Once it got plaintext from the users then the system will generate matching key to decrypt it using the decryption algorithm. And transmit key with OTP to the users who need to decrypt the plaintext from the encrypted data [15].

F. *Decryption*

Firstly, user got the key and OTP to retrieve information from the encrypted which is stored in the cloud. This decryption uses the cipher text c ∈ {0, 1} ^∗ u with the help of key K and nonce N ∈ {0, 1} ^n. the system will search the matching encrypted data with the given key K and OTP from the sender. Once receiver got these details then the proposed system allows him to search content which is stored with the key and OTP. So user can easily collect the data and decrypt the data to get plaintext [16].

III. REQUIREMENTS FOR IMPLEMENTATION

The requirements of hardware and software of the proposed system are as follows:

A. *Software Specifications*

Table 1, shows the software specifications used in developing this application.

Table 1: Software Specifications

Type	Specification
Operating System	Linux- Macintosh
IDE	Android Studio 1.5.1 Build 141.34
Programing Language	Java
Runtime Environment	JRE 8 Update 73
Development Kit	Jdk 1.8.0_51
Database	MySQL 4.1
Crypto Library	Java. Security
Android Simulator	Nexus 5 API 21(Lollipop)

B. *Hardware Specifications*

The hardware specifications of the device used are specified in table 2:

Table 2: Hardware Specifications

Type	Specification
Type of the system	OS X Yosemite
Speed of the Processor	Intel® Core i7 Processor 2.7GHz
Memory Storage	8 GB RAM 1867 MHz DDR3
Graphics	Intel Iris Graphic 6100 1536 MB

Section IV shows step by step procedure of OCB encryption and decryption algorithm for converting the plain text to cipher text and vice versa. General features about the OCB are explained in Section V.

IV. SYSTEM ARCHITECTURE

The below Figure 2, the flow diagram provides detailed information about the proposed data vault system which use OCB algorithm to provide security for the user's confidential data. Firstly, Users need to register with the system by entering their personal details such as name, email address, user name and password to login into the system. Once they complete this process then they will get individual authentication resources such as user name, password to login into the system.

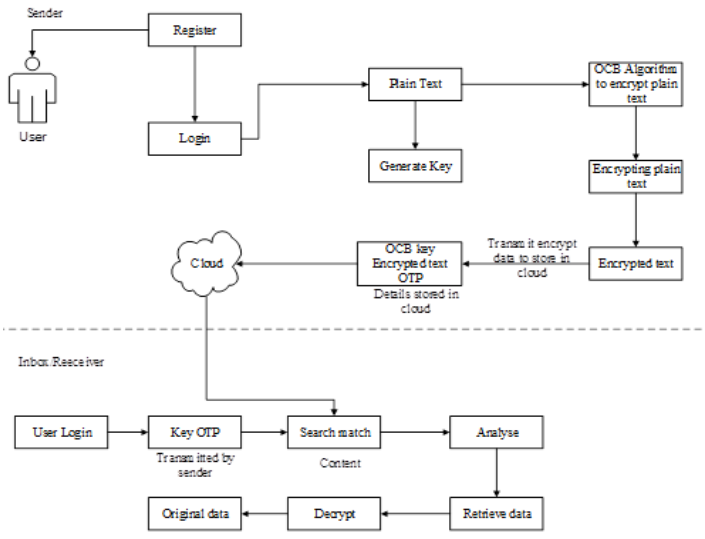

Figure 2: Proposed system architecture

User need to authenticate with the system to use it functionalities once they entered into the system then they have to enter the plaintext which they need to encrypt using the OCB algorithm. Key is generated automatically in the system to encrypt and decrypt plaintext which is entered by

the user. Data is encrypted by the OCB algorithm then it can be transmitted to data receiver with key and encrypted data with OTP.

Data receiver will get the decryption key and OTP to retrieve information from the cloud which is stored by the sender. With the help of key and OTP the system will automatically identify the data from the cloud and decrypt the plaintext.

Receiver will get the key and OTP in their inbox. Using the OTP and Key they can easily search the content stored in the cloud database. The system functionalities will automatically find the data that is stored under the Key and OTP. Once it got the encrypted data then it uses decryption algorithm to retrieve the plaintext from the encrypted data.

C. Data flow diagram of the proposed system

The below figure 3 and figure 4, flow diagram shows the data flow in the proposed from the scratch of how the encryption and decryption is done in the background of the Data Vault android application.

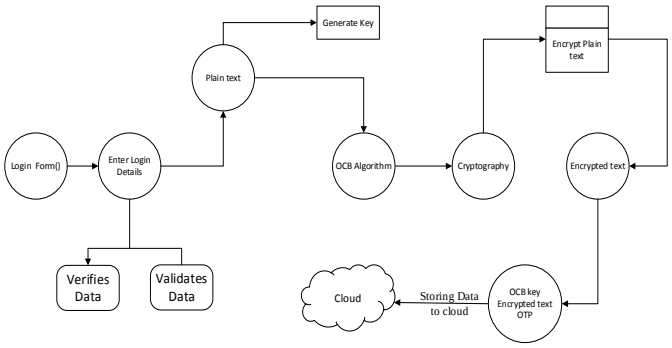

Figure 3: Dataflow diagram of proposed system data encryption

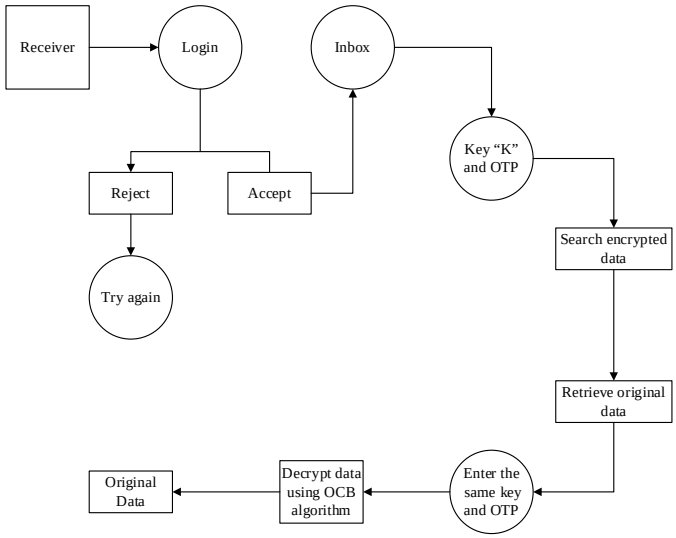

Figure 4: Dataflow diagram of proposed system data decryption.

D. Steps involved in the proposed system

Registration: this process allow customer to register them as a user in the proposed system. By entering customer personal details such as name, email address and password to authenticate in the system. Once user completes the above process they will get user name and password to login into the proposed system.

Login: by entering individual user name and password in the system user can access the system functionalities. This functionality helps to provide security for the system; only authorized user can access the system. To avoid unauthorized user accesses in the system and to avoid some possible risks in the present system this functionality is enabled in the proposed system [9].

Plaintext: once user entered into the system, then the system asks to the user to enter plaintext which needs to encrypt to provide security for the data. This plaintext either numeric or string OCB will encrypt any type of data given by the user. So based on the user need they can encrypt any kind of data using the OCB algorithm.

Generating key: Key is one of the most important aspects in the data encryption and decryption process. The OCB algorithm will automatically generate key K based on the data entered to encrypt in the system. This key is integrated with the OTP and transmitted to the receiver to use it for the decryption.

OCB algorithm to encrypt and decrypt plaintext: OCB mode algorithm is designed for both the encryption and decryption process. In the proposed system OCB use key K and OTP code to encrypt and decrypt the data which is stored in the cloud storage and it provides high security for the user's personal and confidential data [10].

Sending Key and OTP: Once the key K and OTP is generated in the system then using any one data transmitting medium we can send it the receiver inbox.

Storing encrypted data into the cloud: The proposed system automatically stores the encrypted data into the cloud storage using the predefined functionalities.

Receiver searching the matching encrypted data in cloud: Receiver gets key "K" and OTP from the sender. The proposed system allow user to search the encrypted data using the key and OTP. Using these details, the system automatically detects encrypted data from the cloud which match with the key and OTP.

Using key and OTP to decrypt the data: The system use both key "K" and OTP to decrypt plaintext from the encrypted data which is stored in the cloud database. Without, key or OTP receiver cannot retrieve the plaintext from the encrypted data.

V. FINAL OUTCOMES OF PROPOSED SYSTEM

This section provides detailed information about the various processes of the proposed system and the data flow in implemented system. At the same time front end of the developed system is explained detailed in the below sections.

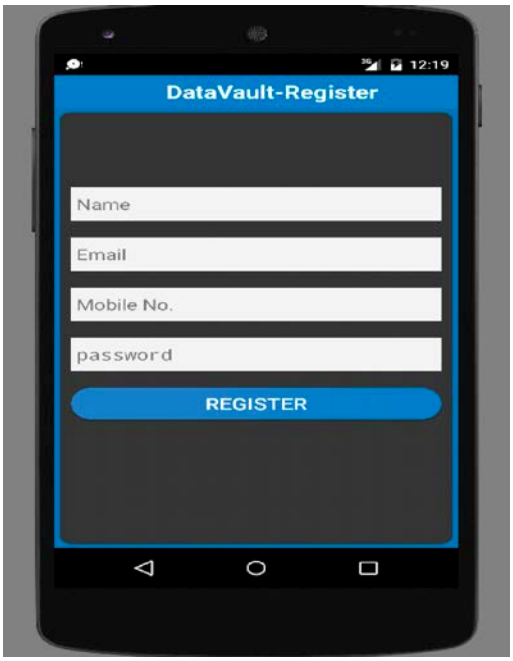

Figure 5: User registration

The proposed system process is categorized into different sections to monitor the activity and flow of the work in the system. Based on the security purpose this proposed system is developed with the user authentication process.

By providing users personal details they will get authentication details such as user name and password in the system. Figure 5 shows the user registration process.

The data vault-login screen (See Figure 6) is used to the user authentication. Each user in the system need to authenticate to use it functionalities without login into the system user cannot enter into the system.

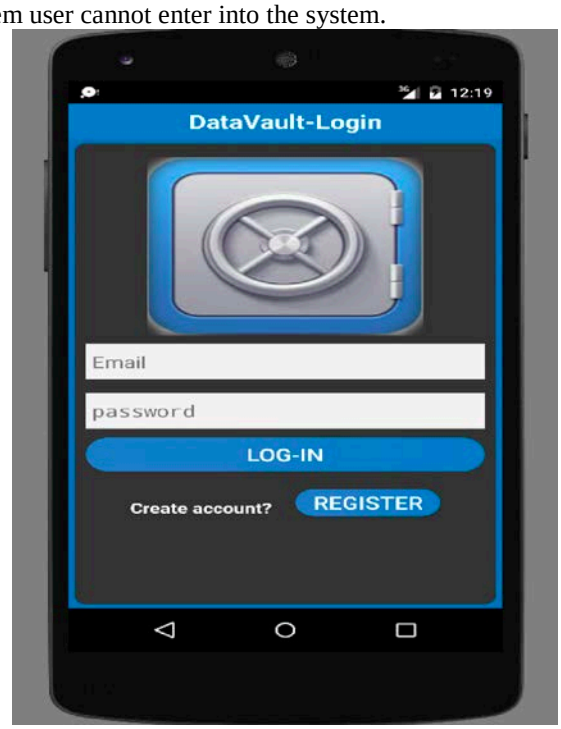

Figure 6: Login screen of the users

Figure 7: The main screen of the proposed system

In this process user (See figure 7) need to enter plaintext which they need to encrypt for providing security to transmit into the receiver in secured manner.

Here, users need to select key, nonce and file size to process encryption in the proposed system. As well as this system provides two functionalities such as send personal details to others in the system and also they view their personal details in the system to edit and updates.

At the same time data vault system users can create three different types of graphs to identify the performance of the proposed system and algorithms. They can create graph to verify algorithm performance in key, nonce and file encryption.

VI. FEATURES OF OCB

OCB algorithm provides various features to its user and to the system as follows;

Fast: Comparing with other algorithms it is very faster as like CTR. Only few XOR's are used in every block encryption on top of an AES call in the system. Anyhow, every chipper text needs average of 1.02 additional AES calls per message in the authentication process of the data encryption and decryption.

Provable secure: OCB is validated many times over a decade of development and research with various papers published in good results. As well it is very secure by using the nonce-based AE scheme with block cipher is a strong PRP in the system to protect data.

Parallel: Many of the OCB computations are done individually by enabling hardware and software acceleration to perform some operation in the functionalities which makes computational units [17] [18].

Timing-attack resistant: The OCB functionalities doesn't have any special conditions and rules to handle these secret data in the process of encryption and decryption.

Online: The OCB process doesn't allow user to view the current state of the data encryption and decryption. It only allows user to view details when it complete all process in the system. Then it automatically starts process based on the user requirement to store data in local or online.

Static AD: If associated data is not converted to plaintext by a series of encryption and the AD contribution is not to be recalculated every time in the encryption process. Anyhow, this process will minimize the successive computation work when multiple encryptions are associated together with the same associated data.

OCB and GCM algorithms have the similar security and features in various settings, but comparing with others OCB is much more efficient in software [19].

Section V highlights the features of OCB authentication encryption algorithm which are provided to the user through the data vault android application. Data Vault android application functions and their screen shots are shown in Section VI.

VII. PERFORMACNE ANALYSIS

Data Vault android application is successfully developed and two main steps are processed in the system they are encrypting the plain text and decrypting the chipper-text. In performance analysis of the proposed system we have taken time to compare which process takes minimum time to complete and which takes more time complete functionalities. Times taken for encrypting and decrypting process are according to the size of the key, nonce and file data. Each files used in this data vault application are in different key and nonce sizes and the time required by this application for encrypting all the data. We can easily identify the best algorithm in comparing its performance in various functionalities. Various key, nonce and file sizes are taken to differentiate the data vault android application system performance. The performances of proposed system with various processes are as follow

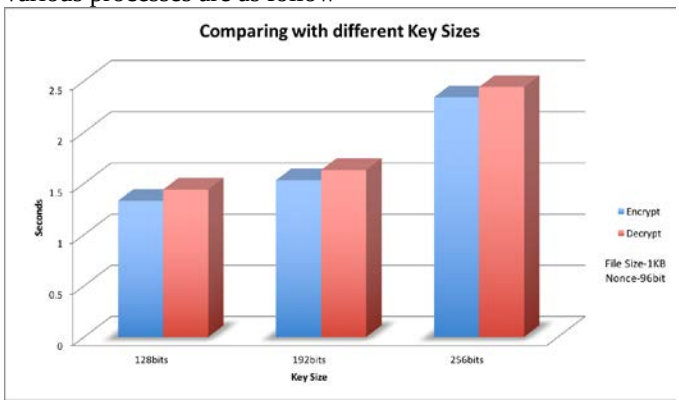

Figure 8: Performance analysis using various key sizes and keeping files size constant – 1kb and nonce 96 bits by varying key sizes

Table 3: Comparing with different key sizes

Key size	128bits	192bits	256bits
Encrypt time (secs)	1.33	1.53	2.34
Decrypt time (secs)	1.44	1.63	2.44

Figure 8, shows the performance of proposed system with various key values. Based on the key size the proposed system algorithm took more time complete both encryption and decryption process. So, using 128-bits key size will be more efficient and takes only minimum time for encrypting and decrypting information in data vault android application.

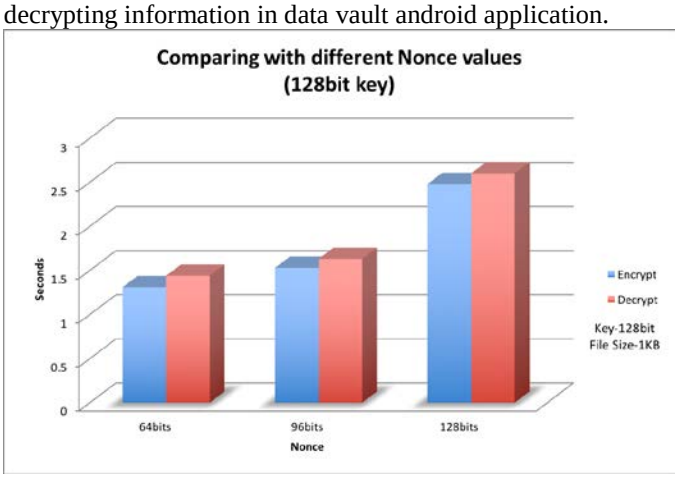

Figure 9: Performance analysis using various nonce values and keeping file size constant – 1kb and key size - 128 bits by varying nonce values.

Table 4: Comparing with different Nonce values

Nonce	64bits	96bits	128bits
Encrypt time (secs)	1.31	1.53	2.48
Decrypt time (secs)	1.44	1.63	2.6

The above Figure 9 provides performance analysis of using various nonce sizes and this analysis also shows using minimum bits will provide a better performance in the data vault android application for both encryption and decryption.

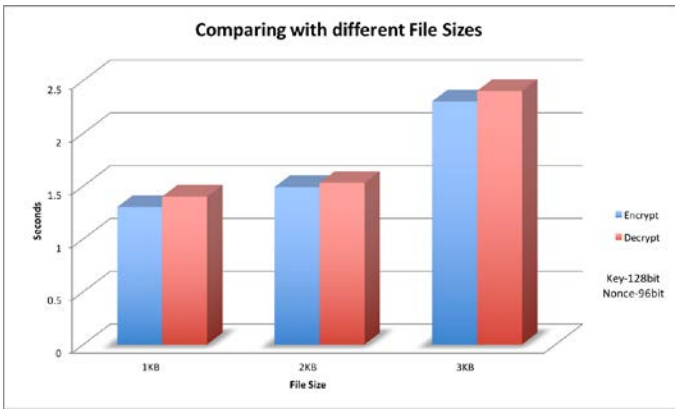

Figure 10: Performance analysis using various file sizes in data vault- android application and keeping nonce value constant – 96 bits and key size - 128 bits by varying file sizes.

Table 5: Comparing with different file sizes

File Size	1KB	2KB	3KB
Encrypt (secs)	1.3	1.49	2.3
Decrypt (secs)	1.4	1.53	2.4

See figure 10, which helps to analyze the performance of data vault-android application in various file size and time to identify which one takes minimum time to complete both encryption and decryption process in the application.

VIII. Conclusion

We present Data vault, android data storage application to store the data safely and securely using OCB mode of authenticated encryption. This application provides confidentiality and privacy to the user data. Authentication process in this application is very efficient, where users need to get authenticated to upload and again have to get OTP authentication to retrieve the data. Cloud data storage is used to store the encrypted data after the OCB transforms the original data into encrypted data. Data vault android application supports the user's sensitive data and adopt the cloud data storage in effective manner. Using this OCB mode algorithm in the system provides more security for its users and this could be a viable solution against all the attackers online. We have also shown that OCB is quite suitable for mobile applications as far as performance is concerned. OCB shows quite acceptable performance level while maintaining required confidentiality and authentication functions in mobile android environments.

IX. References

[1] Bellovin, S. a. (June 2005.). "Guidelines for cryptographic key management". *RFC 4107,*.

[2] Black, J. a. (2002). Side-channel attacks on symmetric encryption schemes: the case for authenticated encryption. *See www.cs.colorado.edu/~jrblack/.*, 327-338.

[3] D. Whiting, R. H. (September 2003.). Counter with CBC-MAC (CCM). *Network Working Group RFC 3610. The Internet Society, .*

[4] Gueron., S. (2009). Intel's New AES instructions for enhanced performance and security. *FSE 2009 LNCS vol. 5665, Springer,* 51–66.

[5] Harkins, D. (October 2008.). "Synthetic Initialization Vector (SIV) authenticated encryption using the Advanced Encryption Standard (AES)". *RFC 5297*.

[6] Iwata, T. (2008). Authenticated encryption mode for beyond the birthday bound security. *AFRICACRYPT 2008, LNCS vol. 5023, Springer,* 125–142.

[7] Krovetz, T. a. (Springer, 2011.). "The software performance ofauthenticated-encryption modes". *in Fast Software Encryption - FSE 2011*.

[8] Lipmaa, H. (July 2001.). Personal communications. *www.tml.hut.fi/~helger.*

[9] McGrew, D. (January 2008). "An Interface and Algorithms for Authenticated Encryption".

[10] McGrew, D. (January 2008.). "An interface and algorithms for authenticated encryption". *RFC 5116*.

[11] N. Ferguson, D. W. (2003). fast encryption and authentication in a single cryptographic primitive. *FSE 2003, LNCS vol. 2887,Springer,* 330–346, .

[12] Rogaway, P. (2004). Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC.

[13] Rogaway, P. (2004). Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. *Advances in Cryptology — ASIACRYPT 2004.*, 16—31.

[14] Rogaway, P. B. (2001). "OCB:A Block-Cipher Mode of Operation for Efficient Authenticated Encryption". *ACM Conference on Computer and Communications Security 2001*.

[15] Rogaway, P. B. (2001). "OCB:a block-cipher mode of operation for efficient authenticated encryption", . *in ACM Conference on Computer and Communications Security 2001 - CCS 2001, ACM Press, 2001*.

[16] Rogaway, P. B. (2003). OCB: A block-cipher mode of operation for efficient authenticated encryption. . *ACM Transactions on Information and System Security* , 365–403 .

[17] Rogaway, P. (Springer, 2004.). "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC". *Advances in Cryptology - ASIACRYPT 2004*.

[18] Rogaway, P. (Springer, 2004.). "Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC". *in Advances in Cryptology - ASIACRYPT 2004*.

[19] Rogaway., P. (May 28, 2012.). OCB Mode. *http://web.cs.ucdavis.edu/~rogaway/ocb/.*

[20] Ted Krovetz, P. R. (July 23, 2012). "The OCB Authenticated-Encryption Algorithm".